Adopting redundancy techniques for multicast stream authentication

T. Cucinotta, G. Cecchetti, G. Ferraro
Scuola Superiore Sant'Anna
P.zza Martiri della Libertà, 33, Pisa, Italy
{cucinotta, gabriele}@sssup.it, gianluca@sasmail.net

Abstract

Various schemes have been proposed to achieve strong authentication of streamed data in a lossy network by means of "light" digital signatures. Such techniques perform a strong authentication on only one packet, to which others are linked by means of hash functions, so that the authentication property propagates to them too. Most of these schemes make the basic assumption that the signature packet is not lost, even if no practical and precise solutions are proposed that guarantee such a property. In this paper we show how adoption of some redundancy techniques can be used in the context of multicast stream authentication in order to increase probability that the signature packets are received and correctly verified against their digital signature. Finally some experimental results are presented comparing computational overheads due to the authentication schemes both at the sender and at the receiver.

This work has been funded by the FABRIC project in the context of the IST Programme of the EU 5th Framework.

Keywords Multicast stream authentication, forward error correction, digital signatures.

1. Introduction

The problem of authenticating digital streams has recently gained more and more importance in the context of multicast transmissions. In fact multicast channels are the main means of delivery of data from a single sender to many receivers or among a group of users. In a multicast setting, the establishment of a group session key allows privacy of the exchanged data and *group authentication*, that is each received packet is guaranteed to come from the user group. It does not achieve, though, *sender authentication*, concerned with the assurance that the data comes from the claimed sender, and *non repudiation*, guaranteeing that the sender cannot deny, at a later moment, to have sent the data. In those applications that have such requirements, like secure delivery of software updates or stock quotes, additional security algorithms must be run to guarantee them.

Basically all of the proposed approaches in the literature make use of *digital signature* schemes to guarantee these security properties. The main problem with adoption of digital signature technology is the high computational overhead due to the complex arithmetics behind today's signature primitives, and the high bandwidth overhead due to the length of a single signature. The computational overhead is a concern especially when transmitters and receivers need to be dedicated, low-cost and low-consumption, dedicated devices. Furthermore, when signing multicast transmissions, many factors must be taken into account, like scalability of the scheme with respect to the size of the multicast group or the usual absence of an upstream channel to return feedback to the sender.

Various schemes have been proposed using "light" digital signature schemes that mainly allow less computational overheads both at the sender and at the receiver sides. Some of these schemes make use of one time digital signatures based on symmetric cryptographic techniques, but this makes impossible to guarantee non repudiation of transmitted data. Other schemes use classic digital signatures based on asymmetric cryptography to authenticate a group of consecutive packets with a single signature, alleviating bandwidth and computational overheads both at the sender and at the receiver. Typically in these approaches a graph of hashes is built on the packet group, where a packet is authenticated at the receiver if and only if the received packets include an oriented path from the packet to the signature. In these works usually there is the assumption that the signature packet is correctly received during transmission, and a little effort has been made to show how this assumption can be validated. Furthermore,performance data on the authentication schemes not always include the additional overhead needed for the preservation of such a property on the signature packet.

In this paper we explicitly analyze how redundancy techniques can be used in digital stream authentication schemes

in order to both guarantee verifiability of the received traffic in presence of an unreliable media and to increase probability that the single signature authenticating a sequence of packets is delivered to the receivers. Furthermore, a prototype of an authenticated multimedia stream server has been implemented in order to gather experimental results about the overall performance impact due to the adoption of these techniques in multicast stream dissemination. The results shown in this paper must be added to the ones pertaining to specific authentication schemes in order to achieve an exact measure of the total overhead due to secure authentication of a digital stream.

1.1. Document structure

The remain of this paper is organized as follows: in section 2 we make a brief survey on multicast strong authentication schemes existing in literature. In section 3.1 we show how general data redundancy techniques can be used to validate the property of correct signature packet delivery for hash chaining based authentication schemes. In section 3 we make a comparison on the performance of some authentication schemes, based on experimental results gathered through prototype implementations. Finally, conclusions are presented in section 4.

2. State of the art

In this section we briefly discuss various methods that have been proposed in last years for providing strong authentication of streams over lossy networks. The discussion is limited to schemes achieving non repudiation of the transmitted data. A simple authentication scheme consists of independently signing each packet of the stream and by including in each packet its digital signature. This scheme suffers of a high computational overhead both at the sender and at the receiver. This is due to the expensiveness of both the sign and the verify operations with current asymmetric cryptographic primitives. Furthermore this scheme achieves a high bandwidth overhead due to the length of usual digital signatures[1].

In many single-sender multi-receiver applications, computational overhead at the sender side is not of great importance, as the sending host can be properly designed to overcome it. Often these applications have much more strict requirements on the receiver side, where low-cost dedicated devices or general purpose home computers must be able to receive and verify the traffic in real time. So many works exist in literature proposing alternative signature schemes with the property of being much less computational expensive, at least for the verify operation.

[1]For a 1024 bit RSA signature, 128 bytes are needed for authenticating each packet

These schemes try to reduce the number of signatures to be performed for a multicast transmission, by allowing a single signature to verify multiple packets at the same time. In this category we have the simple approach in [2], where a single classic digital signature is calculated on the first packet and each packet contains a hash of the next one. This way the authentication performed by the first signature is propagated to the overall stream by means of hash chaining, given that a collision resistant hash function is used. This approach is only suitable when the entire stream is known in advance to the sender and does not tolerate packet losses, as once the chain has been broken due to a lost packet, it is not possible for the receiver to authenticate subsequent received packets anymore.

The general idea of propagating the authentication property by means of authentication chains has been widely investigated and used in the literature in further, more sophisticated, schemes that solved the main drawbacks just cited. In [11] a remarkable work has been made for tolerating arbitrary loss patterns on received packets, by means of two approaches. In the *star-chaining* technique a group of consecutive packets is signed, then the digital signature is included in each transmitted packet, along with the hashes of all the other packets of the group. This way each packet can be verified independently of the others in the group. The main drawback of this method is the quadratic bandwidth overhead with respect to the number of packets in the group. In the more sophisticated approach, the *tree-chaining* technique, a balanced tree of hashes is built, where the leafs of the tree are the hash values of the packets pertaining to a group. Each intermediate node, instead, contains a combination of the hashes of its child nodes. In this case only the root node's hash is digitally signed and included in each transmitted packet, along with the values corresponding to the sibling nodes along the path from the packet to the root. This approach achieves a $n \log n$ logarithmic bandwidth overhead with respect to the sub-stream size, while receiver overhead is still the one of a single signature verification and a linear number of hash computations and comparisons.

A generalization of the simple hash-chaining method has been introduced by Golle and Modadugu in [3], where an optimal solution is presented with respect to available memory resources at the sender and receiver side, in the case of network with losses occurring in bursts. Tolerating burst losses instead of random ones seems to be appropriate for the Internet, as shown by Paxson in [16]. In the Golle construction a complex oriented graph is built, with each edge from a packet to another meaning that the second one includes a hash of the first one. The graph construction starts from an extended chain where each node is connected to the subsequent one, and to the a^{th} subsequent one, for a fixed integer a. Then a method is introduced to insert, between

each subsequent two nodes of the chain, p further nodes. The final construction is somewhat complex and is built in a recursive way. The authors prove resistance of the construction to bursty losses of length at most $p(a-1)$, and its optimality with respect to resources available at the sender and receiver in terms of number of available packet and hash buffers. In this scheme, a packet is authenticated as long as the received packets include an oriented path from the packet itself to the digital signature packet, that needs to be correctly delivered. This last assumption is made throughout all the paper, but only a few hints are given on how to validate it, like multiple retransmissions, while it has not been specified how this would impact the presented performance data.

Another authentication scheme is Efficient Multichained Stream Signature (EMSS) proposed by Perrig et al. in [5]. Using a combination of one-way hash functions and signatures, authentication is obtained by storing the hashes of each packet in multiple locations. Each packet contains a fixed number of hashes from other packets. At the sender side, the actual hashes contained in each packet are calculated at random. The final packet contains the digital signature relative to the group and the hashes of a few last preceding packets. The authors propose to send the signature packet more times in order to guarantee its correct delivery.

Similarly to EMSS, the Augmented Chain [3] uses one-way hash functions and signatures that are periodically sent in the stream. The chain is constructed in two steps: the first one is very similar to EMSS: each hash packet of external chain is stored in the next packet and in the a^{th} following packet, with a fixed parameter a. In the second step, p packets are inserted in the original chain and its hashes are stored in recursive way storing hashes of packet P_i as in previous packets as in subsequent. We sign last packet in the group. This scheme can tolerates burst of $a(p-1)$ packets.

A further scheme is Piggy Backing, introduced in [8]. Main aim of this scheme is the ability to tolerate multiple bursts of a fixed maximum size. In this scheme the group is partitioned in subgroups of packets, called classes. Only the packets in the first class contain hashes: The packets of higher priority classes are spaced by a fixed number of packet from lower priority classes. The first packet in S_0 is signed.

A novel approach has been introduced by Park et al. [13], by adopting Rabin's Information Dispersal Algorithm to construct an authentication scheme that amortizes a group authentication data over all the groups packets. In this approach the packet hashes and the digital signature authenticating the entire group are concatenated, then a redundancy technique is applied on the resulting authentication data in order to split it into pieces through the group packets. If at least a prefixed number m among the n sent packets are received, then the receiver has enough information to reconstruct the entire group authentication data, and authenticate the received packets, as well as the remaining packets of the group that will be received. In the proposed approach, a space-optimal redundancy technique [12] is adopted when splitting the authentication data, making use of Galois Fields in 2^k ($GF(2^k)$). The authors also observe how their construction is highly tolerant to both random and burst loss patterns even in case of high network losses. As we will observe later, the only drawback is on the computational side, where the redundancy technique requires higher resources on the receiver than the graph chaining based approaches do.

Another drawback of this scheme is that it is highly vulnerable to Denial of Service (DoS) attacks, as evidenced by the same authors in [14]. This happens because the receiver, in case of malicious introduction in the stream of bogus packets, has no immediate way for discarding them. Only after receive of m packets, and a FEC decoding computation, the receiver detects a failure in the reconstruction of the authentication data. In the same paper, various techniques are proposed to face with this problem, based upon cryptographic fingerprinting techniques, that allow the receiver to immediately distinguish among original and maliciously introduced packets. An alternative technique for facing with DoS attacks is introduced by Pannetrat et al. [15], who propose sending FEC encoded authentication data relative to a packet group along with the previous packet group.

In the following table, we summarize main properties of the introduced authentication schemes. Specifically, for each of scheme we report the loss patterns the scheme is resistant to, the delay introduced at the sender and receiver sides, measured in number of packets.

Augmented Chain	
resistance to loss	burst of max len $a(p-1)$
sender delay	p
receiver delay	$2n$
Piggy Backing	
resist. to loss (for class i)	x_i bursts of max size b
sender delay	n
receiver delay	n
EMSS	
resistance to loss	variable
sender delay	1
receiver delay	$2n$
SAIDA	
resistance to loss	at most $n-m$ out of n
sender delay	n
receiver delay	n

3. Results

3.1. Guaranteeing delivery of signature

A possibility in guaranteeing correct delivery of the digital signature packet in hash chaining based authentication schemes, usually a simple retransmission has been proposed by authors. An alternative solution could be adopting a FEC encoding scheme for this purpose, like the erasure codes presented in [12]. Figures 1 and 2 report theoretical probability of correct delivery of the signature packet versus the required bandwidth overhead in both cases, at varying single packet loss rates. In both figures only marked points correspond to feasible overhead- probability pairs, though points corresponding to a same loss rate have been connected with lines to facilitate picture reading.

Figures highlight how the FEC based approach allows a much finer granularity in choosing the bandwidth overhead, and that even at low overheads, it reaches higher probability of correct packet delivery, provided that a minimum treshold overhead is used.

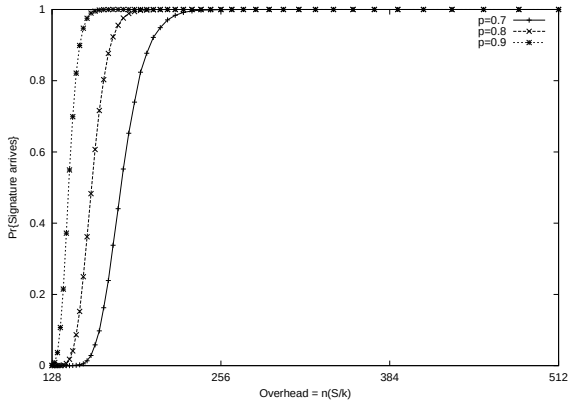

Figure 1. Theoretical overhead vs. target delivery probability with multiple repetitions

3.2. Error Model

A two states, discrete time, Markov Chain has been used to model packet losses during experimentations. According to this model, the communication among the sender and each receiver behaves like a state machine that stops packets when in a state, and lets them through when in the other state. This model produces bursty packet losses, that is appropriate [7] for a number of multimedia network streaming applications. Parameters characterizing the MC are the probability of transition from the normal state to the stopping state, and vice versa. These parameters affect the resulting bursty loss patterns. For the purposes of our experimentation, parameters have been set up so to fix the medium burst length to eight packets, while the loss rate has been varied.

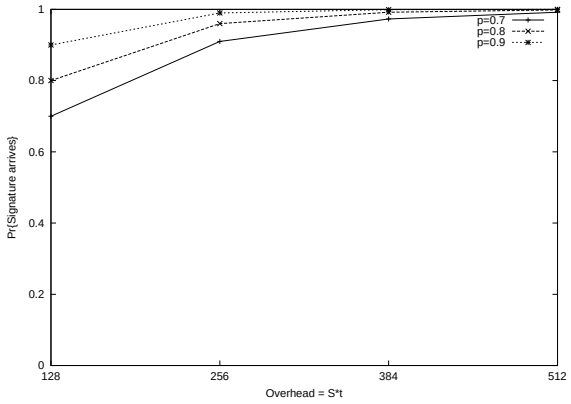

Figure 2. Theoretical overhead vs. target delivery probability with erasure codes

3.3. Experiment set up

Common parameters	
Group size	128 packets
Packet size	1000 bytes
Signature type	1024 bit RSA
Hash type	MD5 (16 bytes)
Medium burst len	8 pkts
Loss rate	5%,10%,20%
EMSS	
Signatures repetitions	0,1,2,3,4
# of hashes in signature	30, 55, 70, 90
Hashes per packet	2, 4, 6
Augmented Chain	
Number of sent signatures	0,1,2,3,4
p parameter	4,8,16
a parameter	2,4,6
SAIDA	
m parameter	32 40 64 70 80 90 100
Piggy Backing	
b parameter	8, 12, 16, 15
Classes	2, 6, 4, 8, 12, 15
Subsequent bursts	1 2 4 6 8

Four authentication schemes have been compared by measuring their performance in a prototype implementation of a multicast stream server. Experiments have been conducted on a 2GHz Athlon processor with FreeBSD operating system.

The considered schemes are: EMSS, PiggyBacking, Augmented Chain and SAIDA. Considered parameters for

the schemes are summarized in the table.

In the first three schemes, in order to guarantee delivery of the signature packet in a lossy environment, it has been transmitted multiple times. Subsequent retransmissions have been distanced in order to avoid their overall loss within a single burst.

3.4. Results

We compared computational overheads due to the authentication of the stream in the four schemes, both at the sender and at the receiver sides. In order to tune up parameters for each scheme, we have performed multiple runs at a fixed loss rate. Only those runs achieving an authentication probability greater than 90% have been considered in the final reports. For each of these cases, we measured the amount of time required by the receiver to decode a single group. Only the time spent in authentication related operations, like signature verification and hash comparisons, has been measured.

Figures 3, 4 and 5 report the encoding times experimented to catch up authentication probabilities greater than 90%, for various loss rates. For parameters resulting in a similar authentication probability, only those achieving the least bandwidth overhead have been reported in the picture.

As the figures show, Augmented Chain and EMSS have the smallest encoding times. SAIDA needs a higher encoding time due to the FEC encoding algorithm. The highest encoding time is obtained for the Piggy Backing scheme.

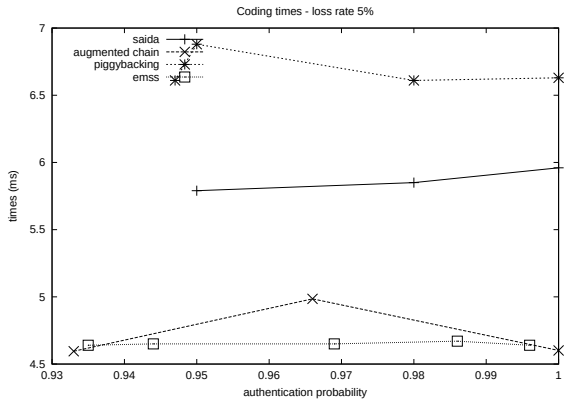

Figure 3. Encoding time vs. achieved authentication probability for a 5% loss rate

Figures 6, 7 and 8 report the decoding times obtained at the receiver in the same cases as for the figures relative to the encoding times.

As the figures highlight, the PiggyBacking, Augmented Chain and EMSS receivers need similar computational time. In fact these schemes all rely on hash calculations

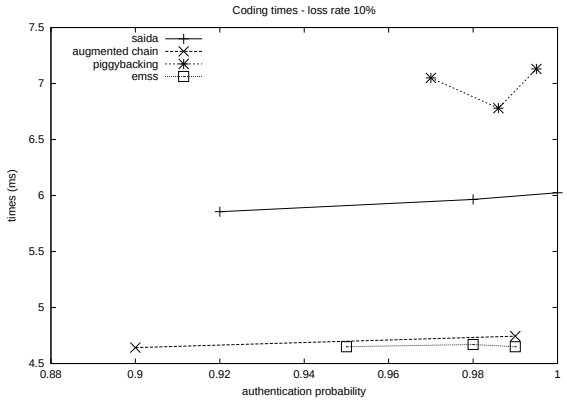

Figure 4. Encoding time vs. achieved authentication probability for a 10% loss rate

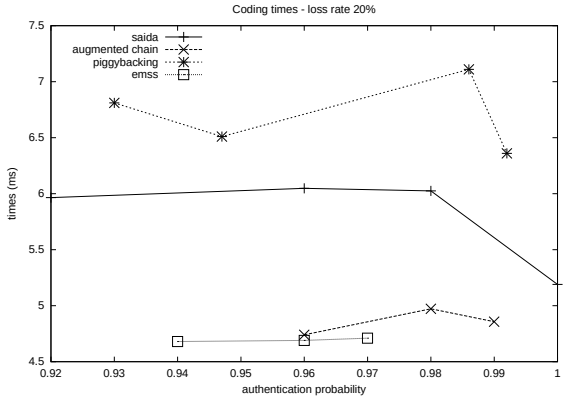

Figure 5. Encoding time vs. achieved authentication probability for a 20% loss rate

and comparisons on the receiver. The SAIDA scheme is more expensive because of FEC decoding operations on the receiver. We must note that the adopted FEC algorithm is a systematic erasure code. So, depending on the actual loss pattern, the experimented decoding time at the receiver has great variations. The best case situation is achieved when all first m packets arrive. In this case, no operations are required to rebuild the authentication data. In the worst case scenario, instead, all first m packets get lost, so, in order to rebuild the authentication data, the receiver must solve a mxm linear system.

Compared with other schemes, SAIDA needs more CPU resources, but the lower bandwidth overhead makes this protocol very promising. Furthermore, among the analyzed schemes, this is one of the simplest one from a point of view of parameter tuning.

Figures 9, 10 and 11 report the needed overhead on the communication due to the adoption of the analyzed authen-

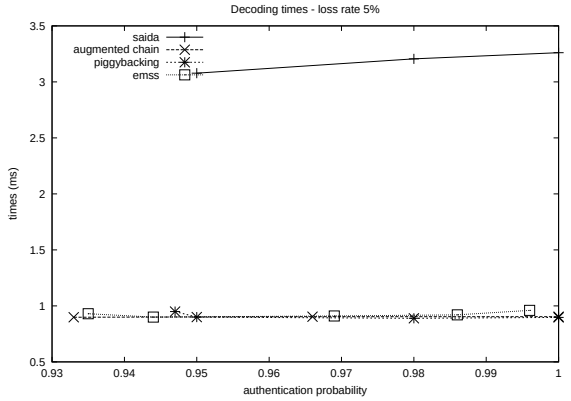

Figure 6. Decoding time vs. achieved authentication probability for a 5% loss rate

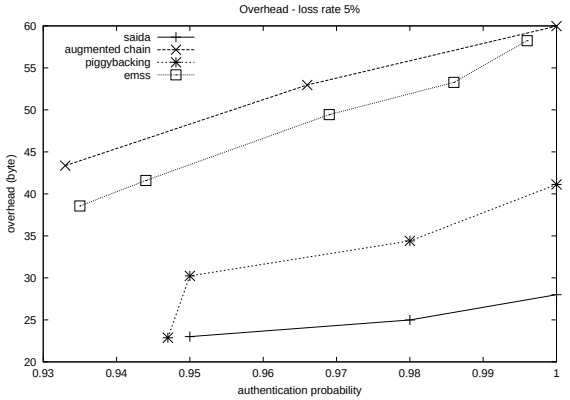

Figure 9. Group overhead vs. achieved authentication probability for a 5% loss rate

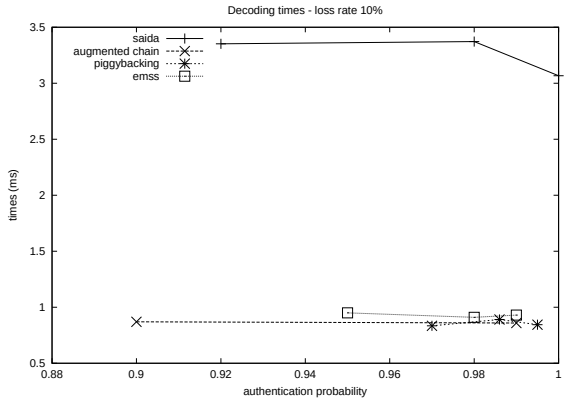

Figure 7. Decoding time vs. achieved authentication probability for a 10% loss rate

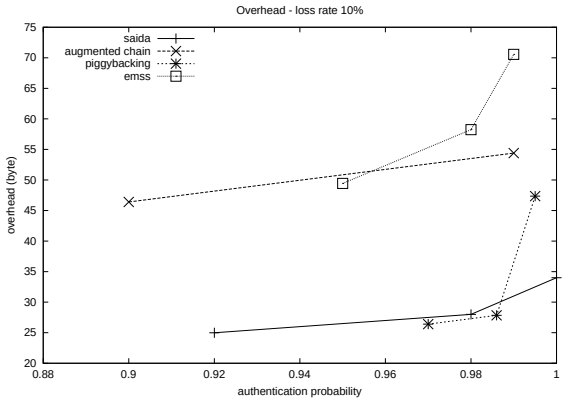

Figure 10. Group overhead vs. achieved authentication probability for a 10% loss rate

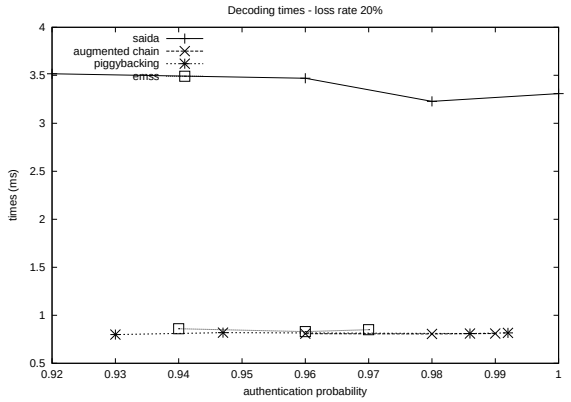

Figure 8. Decoding time vs. achieved authentication probability for a 20% loss rate

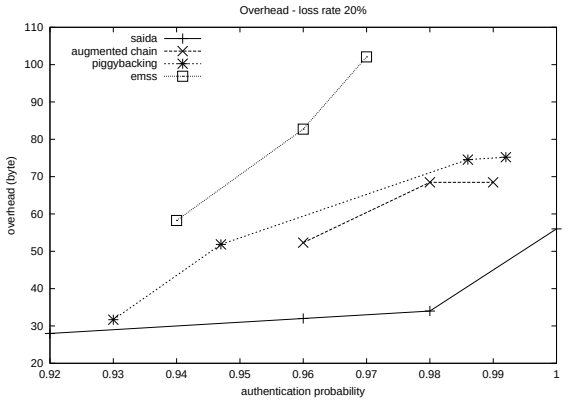

Figure 11. Group overhead vs. achieved authentication probability for a 20% loss rate

tication schemes. The overhead is basically due to the addition of hashes and digital signature information to the original packets. Furthermore the retransmission of the digital signature packet for the EMSS, PiggyBacking and Augmented Chain schemes has been accounted for the final overhead computation, in the exposed results.

As figures show, the SAIDA algorithm requires lower bandwidth overhead than the other algorithms for a target authentication probability.

4. Conclusions

In this paper we made a performance comparison on resource requirements of various multicast authentication schemes. Our attention has been focused on schemes achieving sender authentication and non repudiation in the transmission. The compared schemes are EMSS, Augmented Chain, Piggy Backing and SAIDA. The first three schemes try to preserve authentication data (i.e. packet hashes) by means of their repetition and multiple inclusion in different packets, while the SAIDA algorithm relies on a general redundancy technique for equally distributing the authentication data among packets which it refers to. Gathered performance data highlight that, in spite of a better resistance to packet losses, that is highlighted by a less bandwidth overhead for a same authentication probability, SAIDA presents a higher authentication computational overhead at the receiver.

References

[1] D. Boneh, G. Durfee and M. Franklin. *Lower Bounds for Multicast Message Authentication* - Eurocrypt 2001.

[2] R. Gennaro and P. Rohatgi. *How to Sign Digital Streams.* - CRYPTO 1997.

[3] P. Golle and N. Modadugu. *Authenticating Streamed Data in the Presence of Random Packet Loss.* - ISOC Network and Distributed System Security Symposium 2001.

[4] A. Perrig, R. Canetti, D. Song and J.D. Tygar. *Efficient and Secure Source Authentication for Multicast.* - ISOC Network and Distributed System Security Symposium 2001.

[5] A. Perrig, R. Canetti, D. Song and J.D. Tygar. *Efficient Authentication and Signing of Multicast Streams over Lossy Channels.* - IEEE Symposium on Security and Privacy, May 2000.

[6] P. Rohatgi *A Compact and Fast Hybrid Signature Scheme for Multicast Packet Authentication.* - 6th ACM Conference on Computer and Communication Security 1999.

[7] M. Yajnik, S. Moon, J. Kurose and D. Towsley. *Measurement and Modelling of the Temporal Dependence in Packet Loss.* - IEEE Infocom 1999.

[8] S. Miner, J. Staddon. *Graph-Based Authentication of Digital Streams* - IEEE Symposium on Security and Privacy, May 2002.

[9] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, B Pinkas. *Multicast Security: A Taxonomy and Some Efficient Constructions.* - IEEE Infocom 1999.

[10] S. Even, O. Goldreich, S. Micali. *On-Line/Off-Line Digital Signatures.* - Journal of Cryptology, 1996.

[11] C. K. Wong, S. Lam. *Digital Signatures for Flows and Multicasts.* - IEEE/ACM Transactions on Networking, Vol. 7, No.4, August 1999.

[12] L. Rizzo. *Effective Erasure Codes for Reliable Computer Communication Protocols.* -

[13] J. M. Park, E. K. P. Chong, H. J. Siegel *Efficient Multicast Packet Authentication Using Signature Amortization* - Proceedings of the 2002 IEEE Symposium on Security and Privacy, 2002.

[14] J. M. Park, E. K. P. Chong, H. J. Siegel *Efficient Multicast Packet Authentication Using Erasure Codes* - ACM Transactions on Information and System Security (TISSEC), Volume 6, Number 2, May 2003, in press.

[15] A. Pannetrat, R. Molva *Authenticating Real Time Packet Streams and Multicasts* - Computer Networks & ISDN Systems Journal, 31, April 1999.

[16] V. Paxson. *End-to-End Internet Packet Dynamics.* - IEEE/ACM Transactions on Networking 7(3), pp. 277-292. Proceedings of SIGCOMM 1997.